

# OLLYDBG

## 1. Valore del parametro «CommandLine» passato allo stack all'indirizzo 0040106E

1. **Individuare l'indirizzo:** Carica il malware in OllyDBG e vai all'indirizzo 0040106E.
2. **Analisi della chiamata:** A questo indirizzo, il malware dovrebbe effettuare una chiamata alla funzione `CreateProcess`. Controlla i parametri passati nello stack, in particolare il parametro `CommandLine`.
  - o Seleziona l'indirizzo della chiamata a `CreateProcess`, fai clic con il pulsante destro e scegli "Follow in Dump" -> "CommandLine".
  - o Puoi visualizzare il valore effettivo di `CommandLine` analizzando i valori passati allo stack prima della chiamata della funzione. È spesso il secondo parametro (perché il primo parametro è `lpApplicationName` che può essere NULL).
  - o Il valore verrà mostrato come una stringa o una sequenza di caratteri nello stack.

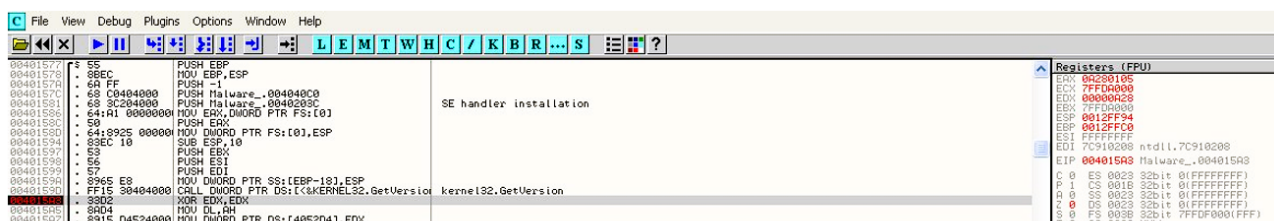
## 2. Inserire un breakpoint software all'indirizzo 00401067 e ottenere il valore di EDX

1. **Imposta un breakpoint:** Vai all'indirizzo 00401067 e inserisci un breakpoint software (F2).
2. **Esecuzione:** Esegui il programma (F9) fino a quando il breakpoint non viene raggiunto.
3. **Valore del registro EDX:** Quando il programma si ferma al breakpoint, controlla il valore del registro EDX visualizzandolo nella finestra dei registri di OllyDBG.

0040105A	. 50	PUSH EAX	pStartupInfo
0040105B	. 6A 00	PUSH 0	CurrentDir = NULL
0040105D	. 6A 00	PUSH 0	pEnvironment = NULL
0040105F	. 6A 00	PUSH 0	CreationFlags = 0
00401061	. 6A 01	PUSH 1	InheritHandles = TRUE
00401063	. 6A 00	PUSH 0	pThreadSecurity = NULL
00401065	. 6A 00	PUSH 0	pProcessSecurity = NULL
00401067	. 68 30504000	PUSH Malware_.00405030	CommandLine = "cmd"
0040106C	. 6A 00	PUSH 0	ModuleFileName = NULL
0040106E	. FF15 04404000	CALL DWORD PTR DS:[&KERNEL32.CreateProcessA]	CreateProcessA
00401074	. 8945 EC	MOV DWORD PTR SS:[EBP-14],EAX	Timeout = INFINITE
00401077	. 6A FF	PUSH -1	hObject
00401079	. 8B4D F0	MOV ECX,DWORD PTR SS:[EBP-10]	WaitForSingleObject
0040107C	. 51	PUSH ECX	
0040107D	. FF15 00404000	CALL DWORD PTR DS:[&KERNEL32.WaitForSingleObject]	
00401083	. 33C0	XOR EAX,EAX	
00401085	. 8BE5	MOV ESP,EBP	

## 3. Eseguire uno step-into e ottenere il nuovo valore di EDX

1. **Step-into:** Esegui un "step-into" (F7) per eseguire l'istruzione corrente.
2. **Nuovo valore di EDX:** Dopo lo "step-into", controlla di nuovo il valore di EDX.

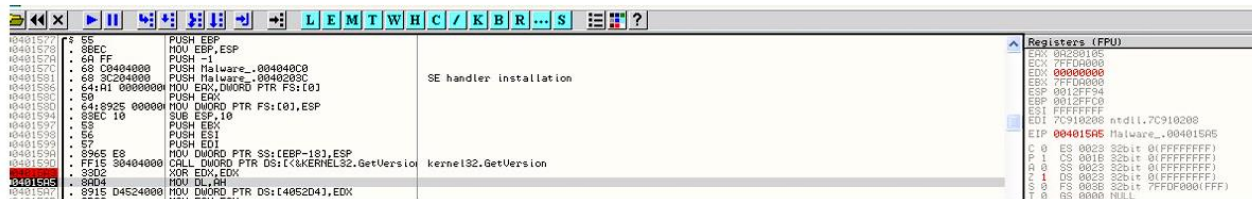


## 4. Motivazione del cambiamento del valore di EDX

- Il valore di EDX può cambiare in base all'istruzione eseguita durante lo "step-into". Se l'istruzione eseguita era una di quelle che modificano il valore di EDX (ad esempio, una MOV, ADD, SUB, XOR o simile), allora EDX sarà aggiornato di conseguenza.

## 5. Identificazione dell'istruzione eseguita durante lo step-into

- Dopo aver eseguito lo step-into, controlla quale istruzione è stata eseguita. Questo può essere fatto osservando l'istruzione in corso nella finestra principale di OllyDBG e comparando l'istruzione appena eseguita con quella che segue.



## 6. Inserire un secondo breakpoint all'indirizzo 004015AF e ottenere il valore di ECX

- Imposta un secondo breakpoint:** Vai all'indirizzo **004015AF** e inserisci un breakpoint software (F2).
- Esecuzione:** Esegui il programma fino a quando il nuovo breakpoint non viene raggiunto.
- Valore di ECX:** Quando il programma si ferma al breakpoint, controlla il valore del registro ECX nella finestra dei registri.



## 7. Eseguire uno step-into e ottenere il nuovo valore di ECX

- Step-into:** Esegui un "step-into" (F7) per eseguire l'istruzione corrente.
- Nuovo valore di ECX:** Dopo lo "step-into", controlla il nuovo valore di ECX 00000005.

## 8. Spiegazione dell'istruzione eseguita

Esadecimale	Binario
0A280105	0000 1010 0010 1000 0000 0001 0000 0101
FF	0000 0000 0000 0000 0000 0000 1111 1111

Eseguendo l'AND logico tra i bit uno ad uno    0000 0000 0000 0000 0000 0000 0000 0101

- **SPIEGAZIONE**
- Come per EDX, l'istruzione eseguita potrebbe essere una che modifica il valore di ECX che in esadecimale è 00000005.